JASPERSOFT

REST API – PHP Demo

Developed by:

 Mariano Luna – Senior Sales Engineer mluna@jaspersoft.com

Published by:

Date: January 15, 2012

License And Copyright Notification

Table of Contents

About REST API Sample

The sample currently provides following functionality of Jaspersoft through the REST API:

- Authenticate to JasperServer
- Browse JasperServer repository
- Run JasperServer reports (HTML, PDF and XLS).
  Only some input control rendering is included in this this version: single select query, multi-select query, single value & boolean.
- Additionally includes a tab that uses the embed theme and IFRAME integration to show adHoc and other features using the simple iframe HTTP integration to include JasperServer content.
 This sample uses the JSESSION ID cookie to authenticate the iframe so there is no exposure of the login credentials in the HTML source code, for this to work properly the application and JRS must be in the same top level domain.

JasperServer REST API upon successful authentication sends a cookie with the JS SESSION ID for use in subsequent REST calls, and the sample uses this in all subsequent calls to keep the user authenticated, if the cookie expires the user will be sent to the login screen.
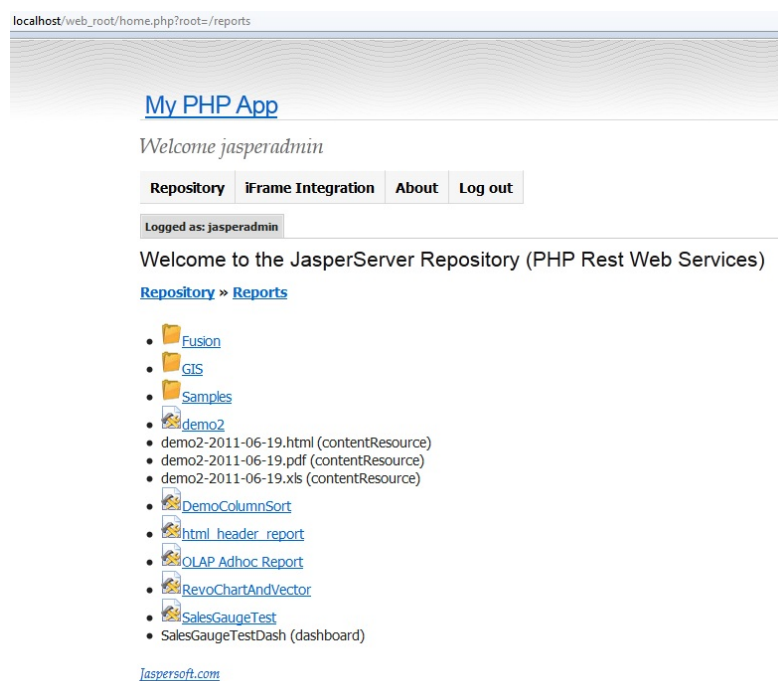
If you have your application running in the same domain as JS (e.g. myexample.com) you can successfully set that cookie in the right path and you do not need to add the authentication (j_username & j_password) to the iframe URL (see the screen shot below).
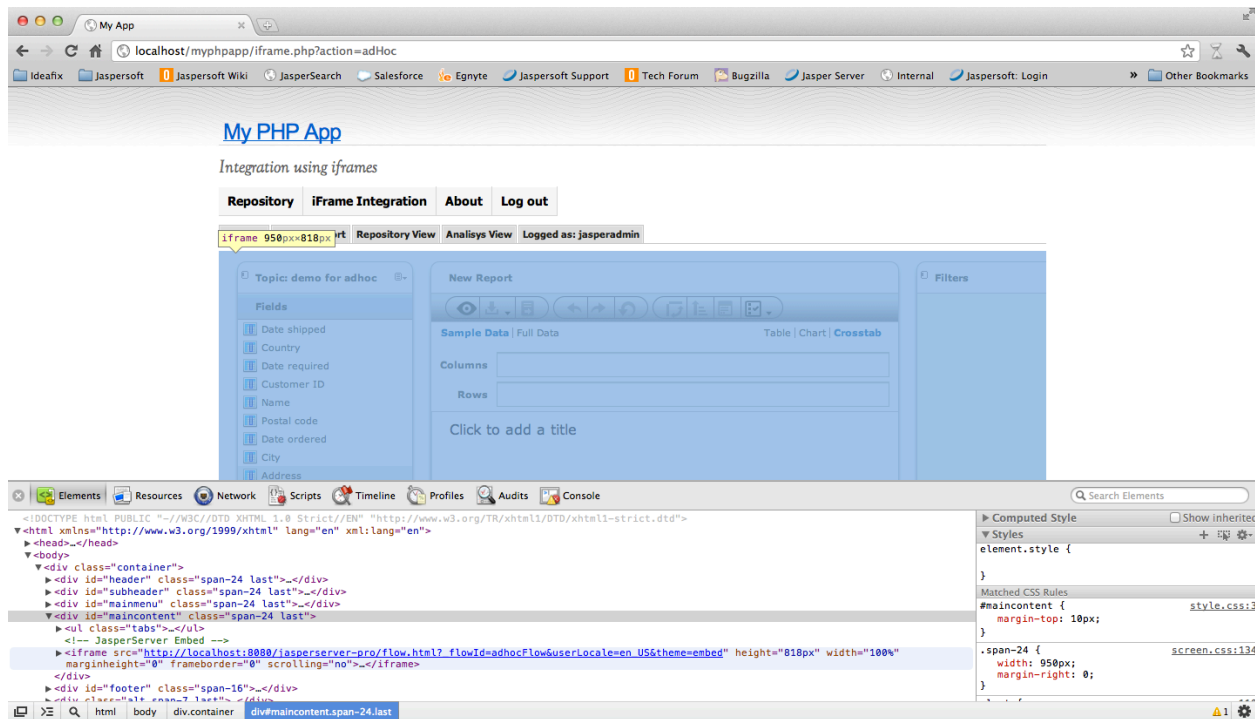
Installation

Requirements

- Installed Jasper Reports Server (JRS) instance on the same top level domain (TLD)
- For iframe integration the "embed" Theme needs to bee installed in your JRS instance. The theme is in *./resources/JS-embed-theme* refer to your Jasper Server documentation for the theme installation.
- A Web Server with PHP support. In this document we are assuming Apache 2.2 and PHP 5.3.x

Install Instructions

Unzip the REST API Sample file in your file system and save that path. For the purpose of this document we will call this path **{SAMPLE_INSTALL_PATH}**, and we will assume:

C:\PHP-Sample

Inside that folder you will find two folders as follows:

1. **./resources**
   This folder contains the included libraries and classes and the main configuration file
   "*config.php*"
2. **./web-root**
   This folder contains the application logic and should be set as your webserver document root.

Apache Configuration

1. As we placed the *web_root* folder outside of Apache standard document root folder, you need to register the folder in the Apache configuration file *httpd.conf*. Consult your apache documentation for the location of the file.
   Add the following code to your apache *httpd.conf* file to mat the sample to the http://servername.com/RESTSample/ URI:

```
---
# Alias for PHP REST APP demo
Alias /RestSample "c:\PHP-Sample\web_root\"
<Directory "c:\PHP-Sample\web_root\">
    Options Indexes  FollowSymLinks MultiViews ExecCGI
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
---
```

   This will create a directory reference and URL alias for the sample app.
2. Is very important to use **AllowOverride All** on this directive since we will be using an.**htaccess** file to modify some of the PHP configuration as we will do below.
3. Modify the **.htaccess** file in the *{SAMPLE_INSTALL_PATH}/web-root* folder:
   a. Edit the file called *htaccess-sample* that is in *{SAMPLE_INSTALL_PATH}/web-root*. There is only one line in this file that overrides the PHP include path to be able to include the libraries in the /resouces folder. Modify this paths to match you installation.
      E.G.:
      For a Windows install they should look like this:
      `php_value include_path "C:/PHP-Sample/resources/;C:/php/PEAR/;."`
      And for Linux / OSX:
      `php_value include_path " /PHP-Sample/resources/:/usr/lib/php/pear/:."`
   b. **SAVE this file as ".htaccess" note the . (dot) at the beginning. (IMPORTANT)**
4. Configuring the application. Edit the file */resources/config.php*
   a. WWW_ROOT (line 48) to match the application root URL, in this case `/RestSample/`
   b. Change the SITE_PATH (line 49) to match the application installation path, in this case `C:\\PHP-Sample\\`
   c. Change the URL to the Jasper Server and Jasper Server Rest client to match youe installation (lines 72 and 79).
5. Restart your Apache Server to reload the new configuration
6. Point your browser to: http://localhost/RESTSample/ and you should see the Sample login page.

7. You can login with your Jasper Server Credentials and start testing the sample.
   If you are using multitenancy your username will be **username|organizationid** (e.g.: jasperadmin|organization_1)

REST API Demo Structure

The iFrame integration is all contained in **iframe.php** and it uses the JRS HTTP API to integrate the full Jasper Server UI in another Web Application.

Most of the logic for the REST API is embed in the following scripts:

- **/web_root/login.php** (authentication)
- **/web_root/home.php** (repository browser)
- **/web_root/viewReport.php** and **/web_root/executeReport.php** (request and execute a report)
- **/resources/RESTclient.php** Constants definition and input control rendering functions.

This REST sample relies on the PEST PHP client (located in **/resources/PEST**) to interact with the JRS RESTful API. The complete PEST project can be found at: https://github.com/educoder/pest/